

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10049368 A**

(43) Date of publication of application: **20.02.98**

(51) Int. Cl

G06F 9/32

(21) Application number: **08200847**

(71) Applicant: **MITSUBISHI ELECTRIC CORP**

(22) Date of filing: **30.07.96**

(72) Inventor: **YAMADA AKIRA
YOSHIDA TOYOHIKO
KENGAKU TOORU**

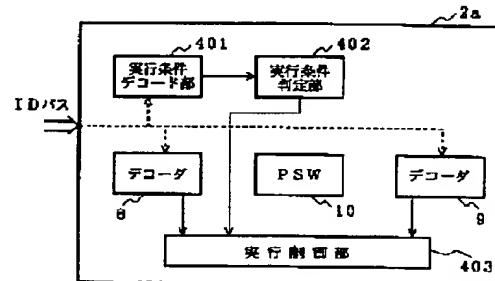
(54) MICROPORCESSOR HAVING CONDITION EXECUTION INSTRUCTION

(57) Abstract:

PROBLEM TO BE SOLVED: To decrease the number of bits in an execution condition field against the condition number by executing an instruction when the coincidence is confirmed between the decoding result of an execution condition code part and the condition that is set by a general-purpose flag.

SOLUTION: An execution condition decoding part 401 extracts an execution condition field of an instruction and decodes the value of extracted three bits into the data corresponding to an execution control flag. If the value of three bits are equal to '000', the relevant code shows that it always executes the instructions. Thereby, the data are outputted to show that the execution control flag is ignored. An execution condition decision part 402 compares these data with the execution control flag. In such cases, the execution control flag is ignored and therefore the part 402 instructs an execution control part 403 to execute the relevant instruction. In response to this instruction, the part 403 outputs a control signal to a memory unit or an integer arithmetic unit.

COPYRIGHT: (C)1998,JPO



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-49368

(43)公開日 平成10年(1998)2月20日

(51)Int.Cl.⁶
G 0 6 F 9/32

識別記号
3 2 0

府内整理番号

F I
G 0 6 F 9/32

3 2 0 F

技術表示箇所

審査請求 未請求 請求項の数4 O.L (全18頁)

(21)出願番号 特願平8-200847

(22)出願日 平成8年(1996)7月30日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 山田 朗

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(72)発明者 吉田 豊彦

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(72)発明者 見学 徹

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

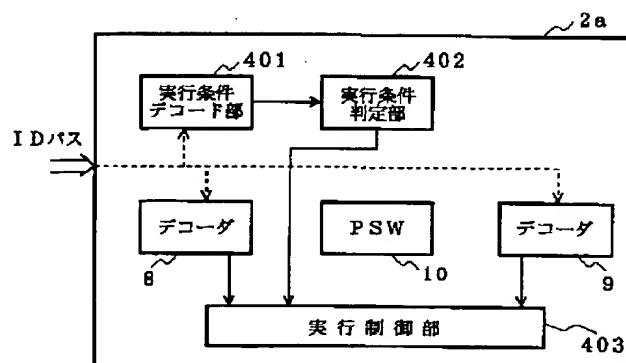
(74)代理人 弁理士 田澤 博昭 (外2名)

(54)【発明の名称】 条件実行命令を有するマイクロプロセッサ

(57)【要約】

【課題】 命令フォーマット中の実行条件フィールドのビット数が多い。

【解決手段】 命令における実行条件フィールドは、命令を実行することを示す汎用フラグの条件がエンコードされた値を有する。命令デコードユニット2は、実行条件フィールド105の値をデコードする実行条件デコード部401と、実行条件デコード部401のデコード結果と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部402とを備える。



【特許請求の範囲】

【請求項1】 実行条件フィールドおよび演算フィールドを含む命令をデコードする命令デコーダと、命令の実行を制御する情報が設定されるレジスタであって条件が設定される汎用フラグを含む制御レジスタと、前記命令デコーダの出力に従って命令を実行する命令実行部とを備えた条件実行命令を有するマイクロプロセッサにおいて、前記実行条件フィールドは、命令を実行することを示す前記汎用フラグの条件がエンコードされた値を有するものであり、前記命令デコーダは、前記実行条件フィールドの値をデコードする実行条件デコード部と、前記実行条件デコード部のデコード結果と前記汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部とを備えたことを特徴とする条件実行命令を有するマイクロプロセッサ。

【請求項2】 実行条件フィールドは、条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部は、前記ビット長のエンコード値をデコードすることを特徴とする請求項1記載の条件実行命令を有するマイクロプロセッサ。

【請求項3】 実行条件フィールドおよび演算フィールドを含む命令をデコードする命令デコーダと、命令の実行を制御する情報が設定されるレジスタであって条件が設定される汎用フラグを含む制御レジスタと、データを記憶する汎用レジスタと、前記命令デコーダの出力に従って命令を実行する命令実行部とを備えた条件実行命令を有するマイクロプロセッサにおいて、前記実行条件フィールドは、命令を実行することを示す前記汎用フラグの条件が設定された前記汎用レジスタ中のレジスタを指定する値を有するものであり、前記命令デコーダは、前記実行条件フィールドで指定されたレジスタを参照するレジスタ参照部と、前記レジスタ参照部が参照したレジスタに設定されている条件と前記汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行することに決定する実行条件判定部とを備えたことを特徴とする条件実行命令を有するマイクロプロセッサ。

【請求項4】 命令デコーダは、1つの実行条件フィールド、1つの命令選択条件フィールドおよび複数の演算フィールドを含む命令をデコードし、命令実行部は、複数の演算回路を有し、前記命令デコーダは、さらに、前記命令選択条件フィールドの設定値に応じて前記複数の演算フィールドによる操作を行うか否かを定める実行制御部を備えたことを特徴とする請求項1から請求項3のうちのいずれか1項記載の条件実行命令を有するマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、命令の一部に含まれる実行条件フィールドを用いて命令の実行を制御する条件実行命令を有するマイクロプロセッサに関するものである。

【0002】

【従来の技術】マイクロプロセッサにおけるパイプライン実行の分岐のペナルティを低減してマイクロプロセッサの性能を向上させるために、条件実行や投機実行などの手法がある。図16は例えばVLSIテクノロジー社の「ARM (ACORN RISC MACHINE) ファミリーデータマニュアル(1990年)」、2-29頁に示された従来のRISCマイクロプロセッサの命令フォーマットの一例を示す説明図である。図に示すように、このマイクロプロセッサにおける条件実行命令は、4ビットの実行条件フィールドを有する。マイクロプロセッサのネガティブフラグ(N)、ゼロフラグ(Z)、キャリーフラグ(C)およびオーバフローフラグ(V)の各状態と実行条件フィールドの一一致/不一致に応じて、命令が実行されるか否か決定される。

【0003】しかし、条件実行の判定に用いられる各フラグは、特定の命令によってのみ設定される専用フラグである。そのために、このようなマイクロプロセッサにおいては、条件実行の範囲が制限される。

【0004】図17は特開平7-182165号公報に示された従来の他のマイクロプロセッサの命令フォーマット300を示す説明図である。図において、301は実行条件が設定される実行条件フィールド、302は操作部分である演算フィールドである。実行条件フィールド301において、303, 304はマイクロプロセッサの内部のフラグ#0に関連するCv0ビット、Cd0ビット、305, 306はマイクロプロセッサの内部のフラグ#1に関連するCv1ビット、Cd1ビット、307, 308はマイクロプロセッサの内部のフラグ#2に関連するCv2ビット、Cd2ビットである。

【0005】図18はCvビットおよびCdビットとそれらが表す意味との関係を示す説明図である。図18において、Cvビットは、Cv0ビット303、Cv1ビット305およびCv2ビット307を代表している。Cdビットは、Cd0ビット304、Cd1ビット306およびCd2ビット308を代表している。図18に示すように、実行条件フィールド301におけるCv0ビット303およびCd0ビット304がともに「1」に設定されている場合には、フラグ#0が真の状態にあるときに命令が実行されうる条件が成立する。Cv0ビット303が「1」に設定されていてCd0ビット304が「0」に設定されている場合には、フラグ#0が偽の状態にあるときに命令が実行されうる条件が成立する。Cv0ビット303が「0」に設定されている場合には、フラグ#0の状態がいずれであっても命令が実行されうる条件が成立する。

【0006】Cv1ビット305およびCd1ビット306

とフラグ#1とについても、図18に示された関係によって、命令が実行されうる条件が成立しているかどうか判断される。Cv2ビット307およびCd2ビット308とフラグ#2とについても、図18に示された関係によって、命令が実行されうる条件が成立しているかどうか判断される。なお、Cvビットの値に応じて、真または偽の判断がなされるのか、それとも判断がなされないかが決まるので、Cvビットは、条件の有効性を判断するためのビットである。また、Cdビットの値に応じて、真であるか偽であるかの判断がなされるので、Cdビットは、値が判断されるビットである。

【0007】次に動作について説明する。マイクロプロセッサは、命令フォーマット300における実行条件フィールド301の設定状態に応じて、演算フィールド302が示す命令を実行するかどうか制御する。例えば、実行条件フィールド301の設定値が「111111」であった場合には、マイクロプロセッサは、フラグ#0, #1, #2が全て真であったときに、演算フィールド302が示す命令を実行する。

【0008】フラグ#0, #1, #2がそれぞれ1ビットで表されているときには、例えば、真は「1」に対応し、偽は「0」に対応する。特開平7-182165号公報に示されたマイクロプロセッサでは、フラグ#0, #1, #2をそれぞれ2ビットで表し、真、偽および未定の3状態を表現する。例えば、フラグ#0, #1, #2中に状態が未定のものがあった場合には、その他の条件が成立していれば、演算フィールド302が示す命令が実行される。そして、実行結果が、一般的のレジスタファイルとは異なるシャドウレジスタファイルに書き込まれる。全ての条件が成立した場合には、その時点で、シャドウレジスタファイルの内容がレジスタファイルに書き込まれる。このようにして、投機実行が実現されている。また、特開平7-182165号公報には、条件数を増やすために、フラグ数がm個 ($m > 3$) の場合も示されている。

【0009】

【発明が解決しようとする課題】従来の条件実行命令を有するマイクロプロセッサは以上のように構成されているので、条件実行の範囲は拡大されるものの、例えば3つのフラグに対して6ビットの実行条件フィールド301が必要とされる。よって、命令フォーマット300中の実行条件フィールド301のビット数が多いという課題があった。

【0010】この発明は上記のような課題を解決するためになされたもので、条件数に対する実行条件フィールドのビット数を低減できる条件実行命令を有するマイクロプロセッサを得ることを目的とする。なお、特開平2-22873号公報には、分岐命令やジャンプ命令を行うか否か判断するための条件のエンコード値がレジスタに設定され、分岐命令やジャンプ命令を行うか否か判断

するための値としてデコード値がレジスタから出力されるものが開示されている。しかし、特開平2-22873号公報には、命令中の実行条件フィールドの構成に関して何等言及されていない。

【0011】

【課題を解決するための手段】請求項1記載の発明に係る条件実行命令を有するマイクロプロセッサは、命令における実行条件フィールドが、命令を実行することを示す汎用フラグの条件がエンコードされた値を有するもの10であり、命令デコーダが、実行条件フィールドの値をデコードする実行条件デコード部と、実行条件デコード部のデコード結果と汎用フラグによる条件とが合致しているか否か判定し合致していた場合に命令を実行することに決定する実行条件判定部とを備えたものである。

【0012】請求項2記載の発明に係る条件実行命令を有するマイクロプロセッサは、実行条件フィールドが、条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部が、そのようなビット長のエンコード値をデコードする構成になっているものである。

【0013】請求項3記載の発明に係る条件実行命令を有するマイクロプロセッサは、実行条件フィールドが、命令を実行することを示す汎用フラグの条件が設定された汎用レジスタ中のレジスタを指定する値を有するものであり、命令デコーダが、実行条件フィールドで指定されたレジスタを参照するレジスタ参照部と、レジスタ参照部が参照したレジスタに設定されている条件と汎用フラグによる条件とが合致しているか否か判定し合致していた場合に命令を実行することに決定する実行条件判定部とを備えたものである。

【0014】請求項4記載の発明に係る条件実行命令を有するマイクロプロセッサは、命令実行部が、複数の演算回路を有し、命令デコーダが、実行条件フィールド、命令選択条件フィールドおよび複数の演算フィールドを含む命令をデコードする構成であって、さらに、命令選択条件フィールドの設定値に応じて複数の演算フィールドによる操作を行うか否か定める実行制御部を有する構成になっているものである。

【0015】

40 【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 図1はこの発明の実施の一形態によるマイクロプロセッサの構成を示すブロック図である。このマイクロプロセッサは、32ビットの内部データバスを有する32ビットマイクロプロセッサである。図において、2は命令RAM6から64ビット幅のIDバスを介して入力した命令コードをデコードする処理を行う命令デコードユニット（命令デコーダ）、3はアドレス計算を行うメモリユニット（命令実行部）、4は論理演算や50シフト演算を行う整数演算ユニット（命令実行部）、5

は32ビット×64ワードの汎用レジスタ、7はデータが格納されるデータRAMである。

【0016】命令デコードユニット2において、8, 9はそれぞれ命令コードをデコードするデコーダ、10はプロセッサの状態を示すプロセッサ状態語(Processor Status Word; 以下、プロセッサ状態語をPSWという)である。命令デコードユニット2は、さらに、デコーダ8のデコード結果とPSW10の内容にもとづいて制御信号11を作成し、それをメモリユニット3に与える。また、命令デコードユニット2は、デコーダ9のデコード結果とPSW10の内容にもとづいて制御信号12を作成し、それを整数演算ユニット4に与える。

【0017】メモリユニット3において、13はジャンプや分岐を含まない命令を実行するとPC(プログラムカウンタ)値に8を加えて次に実行する命令に対するPC値を算出するとともに、ジャンプや分岐を含む命令の実行時に分岐変位をPC値に加算したり、演算で指定されたアドレッシングモードに応じた計算を行ってジャンプ先の命令に対するPC値を計算するPC制御部である。また、PC制御部13は、計算したPC値を32ビット幅のIAバスを介して命令RAM6に送り、命令RAM6から命令コードを出力させる。14はオペランドとなるデータのアクセスを制御するメモリ制御部である。メモリ制御部14は、32ビット幅のDAバスを介してアドレスデータをデータRAM7に転送し命令実行に必要なデータを64ビット幅のDDバスを介してアクセスする。15は汎用レジスタ5から32ビット幅のS1バス、S2バス、S3バスを介して転送された最大3ワードのデータを用いて算術論理演算を行い演算結果を32ビット幅のD1バスを介して汎用レジスタ5に転送するALU、16は汎用レジスタ5からS1バス、S2バス、S3バスを介して転送されたデータを用いてシフト演算を行い演算結果をD1バスを介して汎用レジスタ5に転送するシフタである。

【0018】メモリユニット3に対して、S1バス、S2バス、S3バス、S4バスを介して、32ビット長のデータを一時に4ワード転送することが可能である。従って、例えば、第1のレジスタの内容と第2のレジスタの内容との和でアドレッシングされるメモリの領域に第3のレジスタの内容をストアするとともに、第3のレジスタの内容をストアしたアドレスに所定値を加算して得られる値でアドレッシングされるメモリの領域に第4のレジスタの内容をストアする2ワードストア命令を実現することができる。また、メモリユニット3は、D1バ

コード: フォーマット

FM=00:2命令
01:2命令
10:2命令
11:1命令

スおよびD2バスを介して、メモリユニット3内の2ワードの演算結果またはデータRAM7から転送された2ワードのデータを汎用レジスタ5に転送することができる。

【0019】整数演算ユニット4において、17は汎用レジスタ5から32ビット幅のS4バス、S5バス、S6バスを介して転送された最大3ワードのデータを用いて乗算を行い演算結果を32ビット幅のD2バス、D3バスを介して汎用レジスタ5に転送する乗算器、18は

10乗算の結果を累積加算または累積減算して保持するアキュムレータである。アキュムレータとして、64ビットのものが2本ある。19は汎用レジスタ5からS4バス、S5バス、S6バスを介して転送された最大3ワードのデータを用いて算術論理演算を行い演算結果をD2バス、D3バスを介して汎用レジスタ5に転送するALU、20は汎用レジスタ5からS4バス、S5バス、S6バスを介して転送されたデータを用いてシフト演算を行い演算結果をD2バス、D3バスを介して汎用レジスタ5に転送するシフタである。

20【0020】このマイクロプロセッサでは、汎用レジスタ5から、最大6種類のレジスタ値を読み出すことが可能であって、読み出されたデータは、それぞれ、S1バス、S2バス、S3バス、S4バス、S5バス、S6バスに出力される。また、汎用レジスタ5には、D1バス、D2バス、D3バスを介して最大3種類のレジスタ値を同時に書き込むことが可能である。

【0021】図2はこのマイクロプロセッサの命令フォーマットを示す説明図である。命令フォーマットとして、(a)に示すような1つの命令コードで2つの演算30(operation)を指示する2演算命令のフォーマット101と、(b)に示すような1つの命令コードで1つの演算を指示する1演算命令のフォーマット102がある。2演算命令のフォーマット101には、フィールド103およびフィールド104からなるフォーマットフィールドと、2つの演算フィールド106, 107と、各演算フィールド106, 107に付属する各実行条件フィールド105がある。1演算命令のフォーマット102には、フィールド103およびフィールド104からなるフォーマットフィールドと、フィールド10

408, 109, 110からなる演算フィールドと、演算フィールドに付属する実行条件フィールド105がある。

【0022】フォーマットフィールドは、以下のようない意味を示す。

実行の順番

operation_0	operation_1
第1	第1
第1	第2
第2	第1
第1

ここで、FMは、フィールド103およびフィールド104からなる2ビットの値である。

【0023】FM=00の場合、この命令は2演算命令であることを示す。そして、演算フィールド106で指示されたoperation_0の演算と演算フィールド107で指示されたoperation_1の演算とが、デコード直後のクロックサイクルで並列に実行される。operation_0の演算はメモリユニット3で実行され、operation_1の演算は整数演算ユニット4で実行される。FM=01の場合、この命令は2演算命令であることを示す。そして、operation_0の演算が、デコード直後のクロックサイクルで実行され、operation_1の演算が、operation_0の演算に対して、1クロックサイクル遅れて実行される。FM=10の場合、この命令は2演算命令であることを示す。そして、operation_1の演算が、デコード直後のクロックサイクルで実行され、operation_0の演算が、operation_1の演算に対して、1クロックサイクル遅れて実行される。FM=11の場合、この命令は1演算命令であることを示す。そして、フィールド108, 109, 110からなる演算フィールドで指示された1つの演算がデコード直後のクロックサイクルで実行される。

【0024】実行条件フィールド105は、以下のような意味を持つ。

コード： 実行条件

CC=000： 常時

001 :	F0=真	かつ	F1=無視
010 :	F0=偽	かつ	F1=無視
011 :	F0=無視	かつ	F1=真
100 :	F0=無視	かつ	F1=偽
101 :	F0=真	かつ	F1=真
110 :	F0=真	かつ	F1=偽
111 :	予約済		

【0025】実行条件フィールド105は、マイクロプロセッサの実行コントロールフラグF0, F1の値に応じて、演算フィールド106, 107のoperation_0の演算やoperation_1の演算、およびフィールド108, 109, 110で構成される演算フィールドの演算が有効であるか無効であるか定める。実行コントロールフラグF0, F1については後で説明する。演算が有効であるとは、演算結果がレジスタ、メモリおよびフラグに反映され、その演算による動作の結果が残ることを意味する。演算が無効であるとは、演算結果がレジスタ、メモリおよびフラグに反映されず、あたかも無効演算(NO P)が実行されたかのような動作の結果が残ることを意味する。

【0026】実行条件フィールド105の値CC=000のときには、実行コントロールフラグF0, F1の値にかかわらず常に演算は有効である。CC=001のときには、実行コントロールフラグF0=真のときにのみ演算は有効である。実行コントロールフラグF1の状態

はいずれでもよい。CC=010のときには、実行コントロールフラグF0=偽のときにのみ演算は有効である。実行コントロールフラグF1の状態はいずれでもよい。CC=011のときには、実行コントロールフラグF1=真のときにのみ演算は有効である。実行コントロールフラグF0の状態はいずれでもよい。CC=100のときには、実行コントロールフラグF1=偽のときにのみ演算は有効である。実行コントロールフラグF0の状態はいずれでもよい。CC=101のときには、実行

10 コントロールフラグF0=真かつF1=真のときにのみ演算は有効である。CC=110のときには、実行コントロールフラグF0=真かつF1=偽のときにのみ演算は有効である。CC=111のときの動作は未定義であり、ユーザは、CC=111となる命令を用いることはできない。

【0027】図3は演算フィールドの詳細な内容を示す説明図である。フォーマット111～117は、それぞれ28ビットで表現される短型の演算フィールド106または演算フィールド107によるものである。フォーマット118, 119は、フィールド108, 109, 110で構成される長型の演算フィールドによるものである。

【0028】フォーマット111(Short_M)は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121, 122、レジスタ番号または6ビット長の即値を指定するフィールド123、およびフィールド123がレジスタ番号を示すのか即値を示すのかを指定するフィールド124で構成される。図3に示すように、フィールド124の値Xが30 「00」、「01」または「11」であるときにはフィールド123がレジスタ番号を示していることを示し、「10」であるときには即値を示していることを示す。このフォーマット111は、レジスタ間接アドレッシングのメモリアクセス演算に用いられる。

【0029】フォーマット112(Short_A)は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121, 122、レジスタ番号または6ビット長の即値を指定するフィールド123、およびフィールド123がレジスタ番号を示すのか即値を示すのかを指定するフィールド125で構成される。図3に示すように、フィールド125の値X'が「0」であるときにはフィールド123がレジスタ番号を示していることを示し、「1」であるときには即値を示していることを示す。このフォーマット112は、算術演算、論理演算、シフト演算およびビット演算に用いられる。

【0030】フォーマット113(Short_B1)は、演算内容を指定するフィールド120およびレジスタ番号を指定するフィールド126で構成される。この50 フォーマット113は、レジスタ指定によるジャンプ命

9
令および分岐命令に用いられる。フォーマット114 (Short_B2) は、演算内容を指定するフィールド120および18ビット長のディスプレイスメントのフィールド127で構成される。このフォーマット114は、ジャンプ命令および分岐命令に用いられる。

【0031】フォーマット115 (Short_B3) は、演算内容を指定するフィールド120、レジスタ番号を指定するフィールド121、レジスタ番号または12ビット長の即値を指定するフィールド128、フィールド128がレジスタ番号を示すのか即値を示すのかを指定するフィールド129、およびゼロ判定にもとづいてフィールド121にもとづく条件ジャンプまたは条件分岐を行うか否か指定するフィールド130で構成される。このフォーマット115は、条件ジャンプ命令および条件分岐命令に使用される。

【0032】フォーマット116 (Short_D1) は、演算内容を指定するフィールド120、レジスタ番号を指定するフィールド121、レジスタ番号または12ビット長の即値を指定するフィールド128、フィールド128がレジスタ番号を示すのか即値を示すのかを指定するフィールド129で構成される。このフォーマット116は、条件ジャンプ命令、条件分岐命令およびリピート命令に使用される。フォーマット117 (Short_D2) は、演算内容を指定するフィールド120、レジスタ番号または12ビット長の即値を指定するフィールド128、フィールド128がレジスタ番号を示すのか即値を示すのかを指定するフィールド129、遅延命令(ディレイド命令)に関するフィールド131で構成される。このフォーマット117は、ディレイドジャンプ命令、ディレイド分岐命令およびリピート命令に使用される。

【0033】フォーマット118 (Long1) は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121、122、32ビット長の即値を指定するフィールド132で構成される。このフォーマット118は、複雑な算術演算、大きな即値を用いる算術演算、大きなディスプレイスメント付きレジスタ間接アドレッシングのメモリアクセス演算、大きな変位の分岐演算および絶対番地へのジャンプ命令などに使用される。フォーマット119 (Long2) は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121、122、32ビット長の即値を指定するフィールド132、およびゼロ判定にもとづいてフィールド132にもとづく条件ジャンプまたは条件分岐を行うか否か指定するフィールド133で構成される。このフォーマット119は、大きな分岐変位を持つ条件ジャンプや条件分岐命令に使用される。

【0034】図4はマイクロプロセッサのレジスタ構成を示す説明図である。このマイクロプロセッサは、図4(a)に示すような64本の32ビット長の汎用レジス

タ5、図4(b)に示すような12本の制御レジスタ150、および図4(c)に示すような2本のアキュムレータ18を持つ。R0の汎用レジスタ140の内容は常に0であり、そこへの書き込みは無視される。R62の汎用レジスタは、サブルーチンからの戻り先アドレスが設定されるリンクレジスタである。R63の汎用レジスタは、スタックポインタであり、PSW10のSMフィールドの値に応じてユーザスタックポインタ(SPU)または割り込みスタックポインタ(SPI)として動作する。制御レジスタ150には、プログラムカウンタ151、PSW10、および各種の専用レジスタが含まれる。図3に示すフォーマット112による演算では、64本の汎用レジスタ5のそれぞれを上位16ビットと下位16ビットとに分けてアクセスできる。また、2本のアキュムレータ18を、上位32ビットと下位32ビットとに分けて別々にアクセスできる。

【0035】図5はPSW10の詳細内容を示す説明図である。PSW10の上位16ビットには、スタックポインタを切り替えるためのSMフィールド171、セル20 フデバッグトラップ(SDBT)の検出を示すEAフィールド172、SDBTの許可を指定するDBフィールド173、割り込み許可を指定するIEフィールド174、リピート動作の許可を指定するRPフィールド175、モジュロアドレッシングの許可を指定するMDフィールド176がある。下位16ビットはフラグフィールド180である。フラグフィールド180には8個のフラグがあり、その中のF0フラグ181およびF1フラグ182は演算の有効/無効を指定する。各フラグの値は比較演算や算術演算の結果に応じて変化する。また、30 フラグ初期化演算で初期化したり、フラグ値書き込み演算で任意の値をフラグフィールド180に書き込むことによって変化する。フラグフィールド180の内容は、フラグ値読み出し演算によって読み出される。

【0036】各フラグは、以下のよう意味を有する。
 SM=0 : スタックモード0→SPIを使用
 SM=1 : スタックモード1→SPUを使用
 EA=0 : SDBTを未検出
 EA=1 : SDBTを検出
 DB=0 : SDBTを非許可
 40 DB=1 : SDBTを許可
 IE=0 : 割り込み非許可
 IE=1 : 割り込み許可
 RP=0 : リピートブロック無効
 RP=1 : リピートブロック有効
 MD=0 : モジュロアドレッシング無効
 MD=1 : モジュロアドレッシング有効
 F0 : 汎用フラグ(実行コントロールフラグ)
 F1 : 汎用フラグ(実行コントロールフラグ)
 F2 : 汎用フラグ
 50 F3 : 汎用フラグ

11

F4 (S) : 飽和演算フラグ
 F5 (V) : オーバーフロー フラグ
 F6 (VA) : 累積オーバーフロー フラグ
 F7 (C) : キャリー / ボローフラグ

12

【0037】以下、このマイクロプロセッサの命令一覧を示す。

A. マイクロプロセッサ機能に関する命令

A-1. ロード / ストア命令

LDB : Load one byte to a register with sign extension
 [1バイトロード (符号拡張あり)]
 LDBU : Load one byte to a register with zero extension
 [1バイトロード (ゼロ拡張あり)]
 LDH : Load one half-word to a register with sign extension
 [1ハーフワードロード (符号拡張あり)]
 LDHH : Load one half-word to a register high
 [1ハーフワードロード (レジスタ上位へ)]
 LDHU : Load one half-word to a register with zero extension
 [1ハーフワードロード (ゼロ拡張あり)]
 LDW : Load one word to a register
 [1ワードロード]
 LD2W : Load two words to registers
 [2ワードロード]
 LD4BH : Load four bytes to four half-words in two registers with sign extension
 [4バイトロード (2レジスタへ、符号拡張あり)]
 LD4BHU : Load four bytes to four half-words in two registers with zero extension
 [4バイトロード (2レジスタへ、ゼロ拡張あり)]
 LD2H : Load two half-words to two words in two registers with sign extension
 [2ハーフワードロード (2レジスタへ、符号拡張あり)]
 STB : Store one byte from a register
 [1バイトストア]
 STH : Store one half-word from a register
 [1ハーフワードストア]
 STHH : Store one half-word from a register high
 [1ハーフワードストア (レジスタ上位から)]
 STW : Store one word from a register
 [1ワードストア]
 ST2W : Store two words from registers
 [2ワードストア]
 ST4HB : Store four bytes from four half-words from two registers
 [4バイトストア (2レジスタの4ハーフワードから)]
 ST2H : Store two half-words from two registers
 [2ハーフワードストア (2レジスタから)]
 MODDEC : Decrement a register value by a 5-bits immediate value
 [即値5ビットのデクリメント]
 MODINC : Increment a register value by a 5-bits immediate value
 [即値5ビットのインクリメント]

【0038】

A-2. 転送命令

MVF SYS : Move a control register to a general purpose register

MVTSYS : Move a general purpose register to a control register
 [制御レジスタから汎用レジスタへ]
 MVFACC : Move a word from an accumulator
 [アキュムレータからの1ワード転送]
 MVTACC : Move two general purpose registers to an accumulator
 [2汎用レジスタ内容のアキュムレータへの転送]

【0039】

A-3. 比較命令

CMPcc : Compare [比較]
 cc = EQ (等しい), NE (不等), GT (より大),
 GE (以上), LT (未満), LE (以下),
 PS (ともに正), NG (ともに負)
 CMPUcc : Compare unsigned [比較(符号なし)]
 cc = GT, GE, LT, LE

【0040】 A-4. 最大値／最小値命令

reserved [予約済]

【0041】

A-5. 算術演算命令

ABS : Absolute [絶対値をとる]
 ADD : Add [加算]
 ADDC : Add with carry [加算(キャリー付き)]
 ADDHppp : Add half-word [ハーフワード加算]
 pp = LLL (レジスタ下位、レジスタ下位、レジスタ下位), LLH (レジスタ下位、レジスタ下位、レジスタ上位), LHL, LHH, HLL, HLH, HHL, HHH
 ADDS : Add register Rb with the sign of the third operand
 [レジスタRbに符号を付ける]
 ADDS2H : Add sign to two half-words
 [2ハーフワードに符号を付ける]
 ADD2H : Add two pairs of half-words
 [2ハーフワード同士の加算]
 AVG : Average with rounding towards positive infinity
 [平均をとる]
 AVG2H : Average two pairs of half-words rounding
 towards positive infinity
 [2ハーフワードそれぞれの平均をとる]
 JOINpp : Join two half-words [2ハーフワードの結合]
 pp = LL, LH, HL, HH
 SUB : Subtract [減算]
 SUBB : Subtract with borrow [ボロー付き減算]
 SUBHppp : Subtract half-word [ハーフワードの減算]
 pp = LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH
 SUB2H : Subtract two pairs of half-words
 [2つのハーフワードの減算]

【0042】

A-6. 論理演算命令

AND : logical AND [論理積]
 OR : logical OR [論理和]

15

NOT	: logical NOT [反転]
XOR	: logical exclusive OR [排他的論理和]
AND FG	: logical AND flags [フラグの論理積]
OR FG	: logical OR flags [フラグの論理和]
NOT FG	: logical NOT a flag [フラグの反転]
XOR FG	: logical exclusive OR flags [フラグの排他的論理和]

【0043】

A-7. シフト演算命令

SRA	: Shift right arithmetic [算術右シフト]
SRA2H	: Shift right arithmetic two half-words [2つのハーフワードの算術右シフト]
SRC	: Shift right concatenated registers [レジスタ連鎖右シフト]
SRL	: Shift right logical [論理右シフト]
SRL2H	: Shift right logical two half-words [2つのハーフワードの論理右シフト]
ROT	: Rotate right [右回転]
ROT2H	: Rotate right two half-words [2つのハーフワードの右回転]

【0044】 A-8. ビット操作命令

B CLR	: Clear a bit [ビットクリア]	20 B SET	: Set a bit [ビットセット]
B NOT	: Invert a bit [ビット反転]	BTST	: Test a bit [ビットテスト]

A-9. 分岐命令

BRA	: Branch [分岐]
BRATZR	: Branch if zero [ゼロなら分岐]
BRATNZ	: Branch if not zero [ゼロでないなら分岐]
BSR	: Branch to subroutine [サブルーチンへ分岐]
BSRTZR	: Branch to subroutine if zero [ゼロならサブルーチンへ分岐]
BSRTNZ	: Branch to subroutine if not zero [ゼロでないならサブルーチンへ分岐]
JMP	: Jump [無条件ジャンプ]
JMPTZR	: Jump if zero [ゼロならジャンプ]
JMPTNZ	: Jump if not zero [ゼロでないならジャンプ]
JSR	: Jump to subroutine [サブルーチンへジャンプ]
JSRTZR	: Jump to subroutine if zero [ゼロならサブルーチンへジャンプ]
JSRTNZ	: Jump to subroutine if not zero [ゼロでないならサブルーチンへジャンプ]
NOP	: No Operation [無操作]

[ディレイド分岐、ジャンプ命令に関する命令]

DBRA	D JMP
DBRAI	D JMP I
DBSR	D JSR
DBSRI	D JSRI

【0046】

A-10. OS関連命令

TRAP	: Trap [トラップ]
REIT	: Return from exception, interrupts and traps [例外、割り込み、トラップからのリターン]

【0047】B. DSP機能に関する命令

B-1. 算術操作命令

MUL	: Multiply [乗算]
MULX	: Multiply with extended precision [倍精度乗算]
MULXS	: Multiply and shift to the right by one with extended precision [倍精度乗算および1ビット右シフト]
MULX2H	: Multiply two pairs of half-words with extended precision [2ハーフワードずつの倍精度乗算]
MULHXPp	: Multiply two half-words with extended precision pp=LL, LH, HL, HH [2ハーフワードの倍精度乗算]
MUL2H	: Multiply two pairs of half-words [2ハーフワードずつの乗算]
MACa	: Multiply and add [積和演算] a(アキュムレータ指定)=0, 1
MACSa	: Multiply, shift to the right by one and add a=0, 1 [1ビット右シフト付き積和演算]
MSUBa	: Multiply and subtract [積和(減算)演算] a=0, 1
MSUBSa	: Multiply, shift to the right by one and subtract a=0, 1 [1ビット右シフト付き積和(減算)演算]

[飽和演算に関する命令]

SAT	SATZ
SATHH	SATZ2H
SATHL	SAT2H

【0048】

B-2. リピート命令

REPEAT	: Repeat a block of instructions [命令ブロックの繰り返し]
REPEAT I	: Repeat a block of instructions immediate [命令ブロックの繰り返し(即値指定)]

【0049】図6はマイクロプロセッサの並列2命令実行時のパイプライン動作を示す説明図である。この動作は、命令のフォーマットフィールドの値FM=00のときに実行される。パイプライン190, 195は、命令フェッチステージ191、デコード/アドレス演算ステージ192、実行/メモリアクセスステージ193、およびライトバックステージ194で構成される。並列2命令実行時には、メモリユニット3での実行と整数演算ユニット4での実行とが並列に行われる。図7はマイクロプロセッサのシーケンシャル命令実行時のパイプライン動作を示す説明図である。この動作は、命令のフォーマットフィールドの値FM=01, 10, 11のときに実行される。パイプライン200は、命令フェッチステージ、デコード/アドレス演算ステージ、実行/メモリアクセスステージ、およびライトバックステージで構成されるが、この場合には、メモリユニット3での実行と

整数演算ユニット4での実行とのうちのいずれかが、一時に実行される。

【0050】図8は条件実行を用いるプログラムの一例を示す説明図である。図において、命令i1, i2による命令群B0は、フラグの状態によらず常に実行され

る。命令i1によって、レジスタr1の値に「1」加算した値がレジスタr3に格納される。そして、命令i2によって、レジスタr3とr4の内容が比較され、一致していれば実行コントロールフラグF0がセットされる。命令i3, i4による命令群B1は、実行コントロールフラグF0=1のときに実行される。命令群B1が実行されると、命令i3によって、レジスタr3の値とレジスタr1の値との和が示すメモリアドレスから1ワードのデータがレジスタr6にロードされる。命令i4によって、レジスタr5とr6の内容が比較され、一致していれば実行コントロールフラグF1がセットされ

50

る。命令 i 6, i 7 による命令群 B 3 は、実行コントロールフラグ F 0 = 1 かつ F 1 = 1 のときに実行される。

【0051】命令 i 5 による命令群 B 2 は、実行コントロールフラグ F 0 = 0 のときに実行される。また、命令 i 8 による命令群 B 4 は、実行コントロールフラグ F 0 = 1 かつ F 1 = 0 のときに実行される。命令 i 1 ~ i 8 が順次記述されたプログラムがあった場合に、実行コントロールフラグ F 0, F 1 の値に応じて、図 8 において矢印で示すようにプログラムが実行される。

【0052】図 9 は命令デコードユニット 2a において条件実行を行うための構成を示すブロック図である。図において、401 は命令における実行条件フィールド 105 をデコードする実行条件デコード部、402 はデコードされた値 CC と実行コントロールフラグ F 0, F 1 とを比較する実行条件判定部、403 は実行条件判定部 402 の比較結果に応じて命令の実行を制御する実行制御部である。その他のものは、図 1 に示すものと同じものである。

【0053】次に動作について説明する。図 8 に示す命令 i 1, i 2 において、命令の実行条件フィールド 105 には、「000」が設定されている。実行条件デコード部 401 は、命令における実行条件フィールド 105 を抽出する。そして、抽出した 3 ビットの値 CC を、例えば、実行コントロールフラグ F 0 に対応するデータと実行コントロールフラグ F 1 に対応するデータとにデコードする。3 ビットの値 CC が「000」であった場合には、既に説明したように、そのコードは命令を常時実行することを示している。そこで、実行コントロールフラグ F 0 を無視することを示すデータと実行コントロールフラグ F 1 を無視することを示すデータとを出力する。実行条件判定部 402 は、それらのデータと実行コントロールフラグ F 0, F 1 とを比較する。この場合には、実行コントロールフラグ F 0, F 1 は無視されるので、実行条件判定部 402 は、実行制御部 403 に対して、その命令を実行するように指示を出す。実行制御部 403 は、その指示に応じて、制御信号 11, 12 をメモリユニット 3 または整数演算ユニット 4 に出力する。

【0054】例えば、命令群 B 3 の命令 i 6, i 7 において、実行条件フィールド 105 には、「101」が設定されている。実行条件デコード部 401 は、3 ビットの値 CC を実行コントロールフラグ F 0 に対応するデータと実行コントロールフラグ F 1 に対応するデータとにデコードする。既に説明したように、そのコードは実行コントロールフラグ F 0 = 真で実行コントロールフラグ F 1 = 真のときに命令を実行することを示している。ここでは、真 = 1 とする。実行条件デコード部 401 は、実行条件として、実行コントロールフラグ F 0 = 1 を示すデータと実行コントロールフラグ F 1 = 1 を示すデータとを出力する。実行条件判定部 402 は、それらのデータと実行コントロールフラグ F 0, F 1 とを比較す

る。そして、実際に実行コントロールフラグ F 0 = 1 かつ実行コントロールフラグ F 1 = 1 であったときには、実行制御部 403 に対して、その命令を実行するように指示を出す。

【0055】図 10 は実行コントロールフラグを F 0, F 1, F 2 の 3 個とした場合の実行条件を示す説明図である。図に示すように、3³ 個の条件が考えられる。全ての条件の指定を可能にするには、デコードされた値 CC として 5 ビット必要である。すなわち、命令における実行条件フィールド 105 は 5 ビット長となる。例えば、CC = 「00000」の場合には、実行コントロールフラグ F 0 = 真、F 1 = 真、F 2 = 真のときにその命令が実行される。

【0056】この場合にも、実行条件デコード部 401 は、命令における実行条件フィールド 105 を抽出する。そして、抽出した 5 ビットの値 CC を、例えば、実行コントロールフラグ F 0 に対応するデータ、実行コントロールフラグ F 1 に対応するデータおよび実行コントロールフラグ F 2 に対応するデータにデコードする。実行条件判定部 402 は、それらのデータと実行コントロールフラグ F 0, F 1, F 2 とを比較し、それらのデータと実際の実行コントロールフラグ F 0, F 1, F 2 の状態とが一致したときに、実行制御部 403 に対して、その命令を実行するように指示を出す。

【0057】以上のように、実行条件フィールド 105 は、条件実行のための条件をエンコードして保持している。従って、3 つの実行コントロールフラグ F 0, F 1, F 2 のすべての組み合わせに対応した条件を 5 ビットの実行条件フィールド 105 で指定できる。図 17 に示す従来の例では、3³ 個の条件を指定するのに 6 ビットの実行条件フィールド 301 が必要であったが、この実施の形態によれば、1 ビット少ない実行条件フィールド 105 で同様の条件実行を実現できる。よって、演算フィールドに割り当てるビット数を増加でき、マイクロプロセッサで提供できる命令数を増やすことができる。

【0058】実行コントロールフラグ F 0, F 1 の数が 2 個の場合に、条件（真、偽、無視）の組み合わせは実は 3² = 9 である。図 2 に示された命令フォーマットの

説明において、3 ビットの実行条件フィールド 105 を示したが、9 個の条件を全て実現しようとすると、実は 4 ビットの実行条件フィールド 105 が必要である。しかし、実行条件フィールド 105 のビット長を長くして全ての条件をユーザに提供することよりも、条件を制限して実行条件フィールド 105 のビット長を短くし、その結果、演算フィールドに割り当てるビット数を増加する方が、ユーザにとって便利であることも考えられる。

【0059】そこで、既に説明したように、実行コントロールフラグ F 0, F 1 の数が 2 個の場合でも、3 ビッ

トの実行条件フィールド105を定義した。図11における実線で示された各組み合わせは、全組み合わせ9個のうち、破線で示された代用可能な組み合わせを除いた7種類である。ユーザは、破線で示された実行コントロールフラグF0=偽、F1=真の条件、および実行コントロールフラグF0=偽、F1=偽の条件を、他の条件、例えば、実行コントロールフラグF0=真、F1=偽の条件、および実行コントロールフラグF0=真、F1=真の条件で代用することになる。図2に示された命令フォーマットの説明において示された7種類の条件は、以上のような考え方にもとづいて定義されたものである。

【0060】以上のように、実行条件の種類を制限すれば、演算フィールドに割り当てるビット数をさらに増加することができる。

【0061】実施の形態2. 図12はこの発明の実施の形態2によるマイクロプロセッサにおける命令デコードユニット2bにおいて条件実行を行うための構成を示すブロック図である。図において、404は命令中の3ビットの実行条件フィールド105の値にもとづいて該当レジスタを参照するレジスタ参照部、406は参照されたレジスタの設定にもとづいて実行条件を判定する実行条件判定部である。その他のものは図9に示したものと同じものである。

【0062】次に動作について説明する。この場合、命令中の実行条件フィールド105は、以下のような意味を持つ。

コード： 実行条件

CC=000 :	常時
001 :	R1で指定
010 :	R2で指定
011 :	R3で指定
100 :	R4で指定
101 :	R5で指定
110 :	R6で指定
111 :	予約済

すなわち、実行条件フィールド105は、汎用レジスタ5を用いて間接的に実行条件を指定する。

【0063】図13はフラグの組み合わせ条件を指定するレジスタ240の内容を示す説明図である。レジスタ240中の各ビットによる実行条件は、図18に示された条件と同じである。この場合、Cv0ビット241、Cd0ビット242は、実行コントロールフラグF0との比較に用いられるビットである。Cv1ビット243、Cd1ビット244は、実行コントロールフラグF1との比較に用いられるビットである。Cv2ビット245、Cd2ビ

CCによる条件不成立のとき：Sビット無視 (operation_0もoperation_1も実行しない)

CCによる条件成立のとき : Sビット=0ならばoperation_0、operation_1ともに実行する

ット246は、実行コントロールフラグF2との比較に用いられるビットである。

【0064】例えば、実行条件フィールド105に設定された値CC='001'の場合には、レジスタ参照部404は、レジスタR1のCv0ビット241、Cd0ビット242、Cv1ビット243、Cd1ビット244、Cv2ビット245、Cd2ビット246を入力する。そして、それを実行条件判定部406に出力する。実行条件判定部406は、Cv0ビット241およびCd0ビット242による意味、Cv1ビット243およびCd1ビット244による意味、Cv2ビット245およびCd2ビット246による意味を、それぞれ、実行コントロールフラグF0の状態、実行コントロールフラグF1の状態、実行コントロールフラグF2の状態と比較する。各意味と各状態とが全て一致していたら、実行条件判定部406は、実行制御部403に対して、その命令を実行するように指示を出す。

【0065】以上のように、この実施の形態2によれば、実行条件フィールド105に設定されたレジスタを指定する値に応じて該当レジスタを参照し、レジスタに設定されている値に応じて条件実行するので、やはり、実行条件フィールド105のビット数を低減することができ、演算フィールドに割り当てるビット数を増加することができる。

【0066】実施の形態3. 図14はこの発明の実施の形態3によるマイクロプロセッサにおける2演算命令のフォーマットを示す説明図である。図に示すように、このフォーマット250には、フィールド103とフィールド104とからなるフォーマットフィールド、2つの

30 演算フィールド106、107、1つの実行条件フィールド105、および1つの命令選択条件フィールド251がある。図15は実施の形態3によるマイクロプロセッサにおける命令デコードユニット2cにおいて条件実行を行うための構成を示すブロック図である。図において、405は命令選択条件フィールド251の内容と実行条件判定部402からの指示とに応じて命令の実行を制御する実行制御部である。その他のものは図9に示されたものと同じものである。

【0067】次に動作について説明する。図14に示す40 フォーマット250における命令選択条件フィールド251は、演算フィールド106によるoperation_0を実行するか演算フィールド107によるoperation_1を実行するか選択するための1ビットのSビットを含む。Sビットは、以下のように用いられる。

S ビット = 1 ならば operation_0 を実行し、operation_1 を実行しない

【0068】 実行条件デコード部 401 および実行条件判定部 402 は、実施の形態 1 の場合と同様に動作する。そして、実行条件フィールド 105 の値 CC による条件が成立した場合には、実行制御部 405 に対して、命令を実行するに指示を与える。実行制御部 405 は、その指示を受けると、S ビットの判定を行う。そして、S ビット = 0 ならば、メモリユニット 3 および整数演算ユニット 4 に制御信号 11, 12 を出力する。S ビット = 1 ならば、メモリユニット 3 に制御信号 11 を出力する。

【0069】 この実施の形態 3 によれば、命令選択条件フィールド 251 によって、operation_0 の実行および operation_1 の実行を制御することができる。すなわち、2 演算命令において、2 つの演算を条件付きで実行することができる。なお、この実施の形態 3 では、実施の形態 1 に対して S ビットの判定処理が追加されたものを見たが、実施の形態 2 に対して S ビットの判定処理を追加してもよい。

【0070】

【発明の効果】 以上のように、請求項 1 記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドの値のデコード結果と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行するように構成したので、実行条件フィールドに割り当てられるビット長を短くでき、その結果、演算フィールドに割り当てられるビット数を増加でき、マイクロプロセッサにおいて提供できる命令数を増やすことができる効果がある。

【0071】 請求項 2 記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドが条件実行の判定に用いられる汎用フラグの全ての組み合わせを表現するビット数よりも少ないビット長を有し、実行条件デコード部がそのようなビット長のエンコード値をデコードするように構成したので、実行条件フィールドに割り当てられるビット長をさらに短くできる効果がある。

【0072】 請求項 3 記載の発明によれば、条件実行命令を有するマイクロプロセッサを、実行条件フィールドで指定されたレジスタに設定されている条件と汎用フラグによる条件とが合致しているか否か判定し、合致していた場合に命令を実行するように構成したので、実行条件フィールドに割り当てられるビット長を短くでき、その結果、演算フィールドに割り当てられるビット数を増加でき、マイクロプロセッサにおいて提供できる命令数を増やすことができる効果がある。

【0073】 請求項 4 記載の発明によれば、条件実行命令を有するマイクロプロセッサを、命令中の命令選択条件フィールドの設定値に応じて複数の演算フィールドに

よる操作を行うか否か定めるように構成したので、2 演算命令において、2 つの演算を条件付きで実行することができる効果がある。

【図面の簡単な説明】

【図 1】 この発明の実施の一形態によるマイクロプロセッサの構成を示すブロック図である。

【図 2】 マイクロプロセッサの命令フォーマットを示す説明図である。

【図 3】 演算フィールドの詳細な内容を示す説明図である。

【図 4】 マイクロプロセッサのレジスタ構成を示す説明図である。

【図 5】 P SW の詳細な内容を示す説明図である。

【図 6】 マイクロプロセッサの並列 2 命令実行時のパイプライン動作を示す説明図である。

【図 7】 マイクロプロセッサのシーケンシャル命令実行時のパイプライン動作を示す説明図である。

【図 8】 条件実行を用いるプログラムの一例を示す説明図である。

【図 9】 この発明の実施の形態 1 における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図 10】 実行コントロールフラグを 3 個とした場合の実行条件を示す説明図である。

【図 11】 実行コントロールフラグが 2 個の場合の実行条件の設定の仕方を示す説明図である。

【図 12】 この発明の実施の形態 2 における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図 13】 フラグの組み合わせ条件を指定するレジスタの内容を示す説明図である。

【図 14】 この発明の実施の形態 3 によるマイクロプロセッサにおける 2 演算命令のフォーマットを示す説明図である。

【図 15】 この発明の実施の形態 3 における命令デコードユニットにおいて条件実行を行うための構成を示すブロック図である。

【図 16】 従来の RISC マイクロプロセッサの命令フォーマットの一例を示す説明図である。

【図 17】 従来の他のマイクロプロセッサの命令フォーマットを示す説明図である。

【図 18】 Cv ビットおよび Cd ビットとそれらが表す意味との関係を示す説明図である。

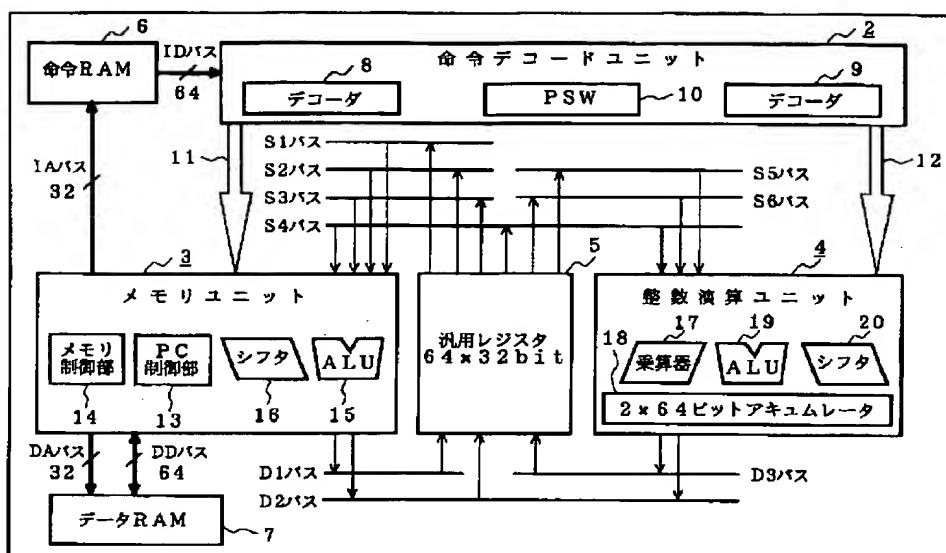
【符号の説明】

2 命令デコードユニット（命令デコーダ）、3 メモリユニット（命令実行部）、4 整数演算ユニット（命令実行部）、150 制御レジスタ、401 実行条件デ

50 コード部、402, 406 実行条件判定部、404

レジスタ参照部、405 実行制御部。

【図1】

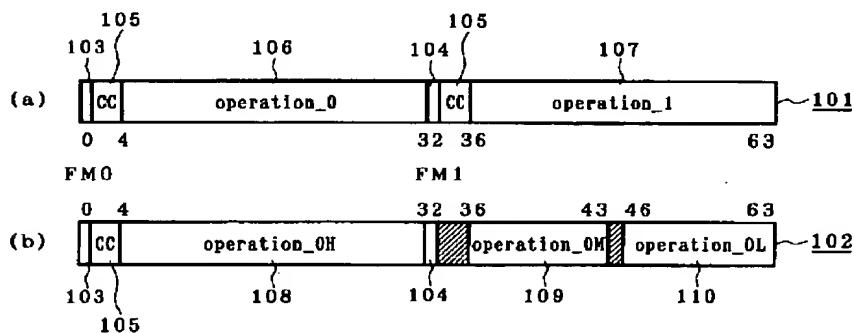


2:命令デコードユニット(命令デコーダ) 3:メモリユニット(命令実行部) 4:整数演算ユニット(命令実行部)

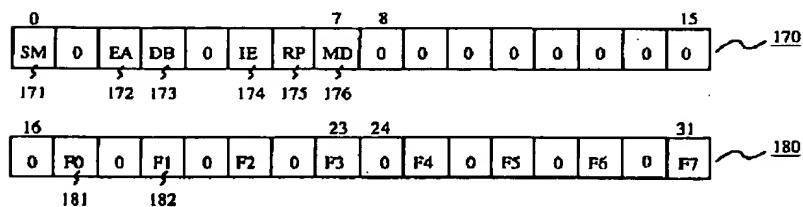
【図18】

有効性Cv	値Cd	意味
1	1	真
1	0	偽
0	x	無視

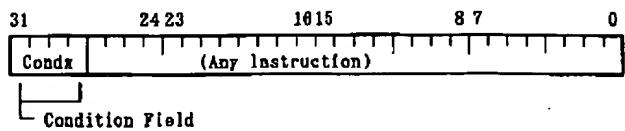
【図2】



【図5】



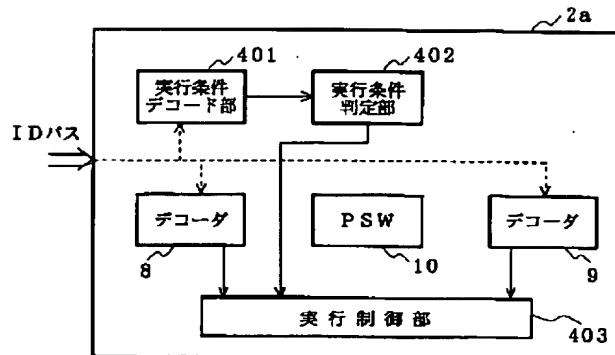
【図16】



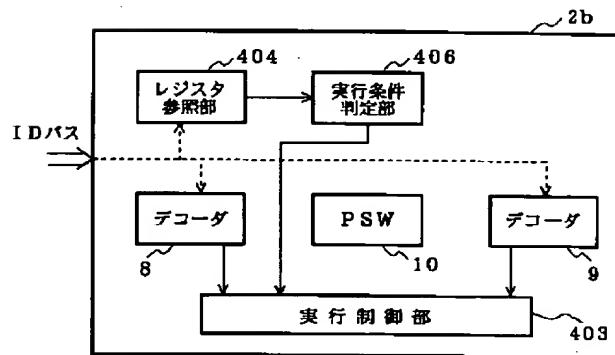
【図3】

	0	7	9	15	21	27	
Short_X	#コート	X	Ra	Rb	Y-X		111
	120	124	121	122	123		(123)
	0	7	9	15	21	27	
Short_A	#コート	X'0	Ra	Rb	Y-X		112
	120	125	121	122	123		(123)
	0	7	9	15	21	27	
Short_B1	#コート	000	0	0	Rc		113
	120				128		
	0	7	9		27		
Short_B2	#コート	10	デイスプ	レイズメント:18			114
	120				127		(128)
	0	7	9	15	27		
Short_B3	#コート	YZ	Ra	Y-X			115
	120	129	130	121	128		
	0	7	9	15	27		
Short_D1	#コート	Y0	Ra	Y-X			116
	120	129	121	128			
	0	7	9	15	27		
Short_D2	#コート	Y0 ct:6	ソース				117
	120	129	131	128			
	0	7	9	15	21		118 53
Long 1	#コート	10	Ra	Rb		即値: 32	
	120				132		
	0	7	9	15	21		119 53
Long 2	#コート	1Z	Ra	Rb		即値: 32	
	120	133	121	122			
					132		

【図9】



【図12】



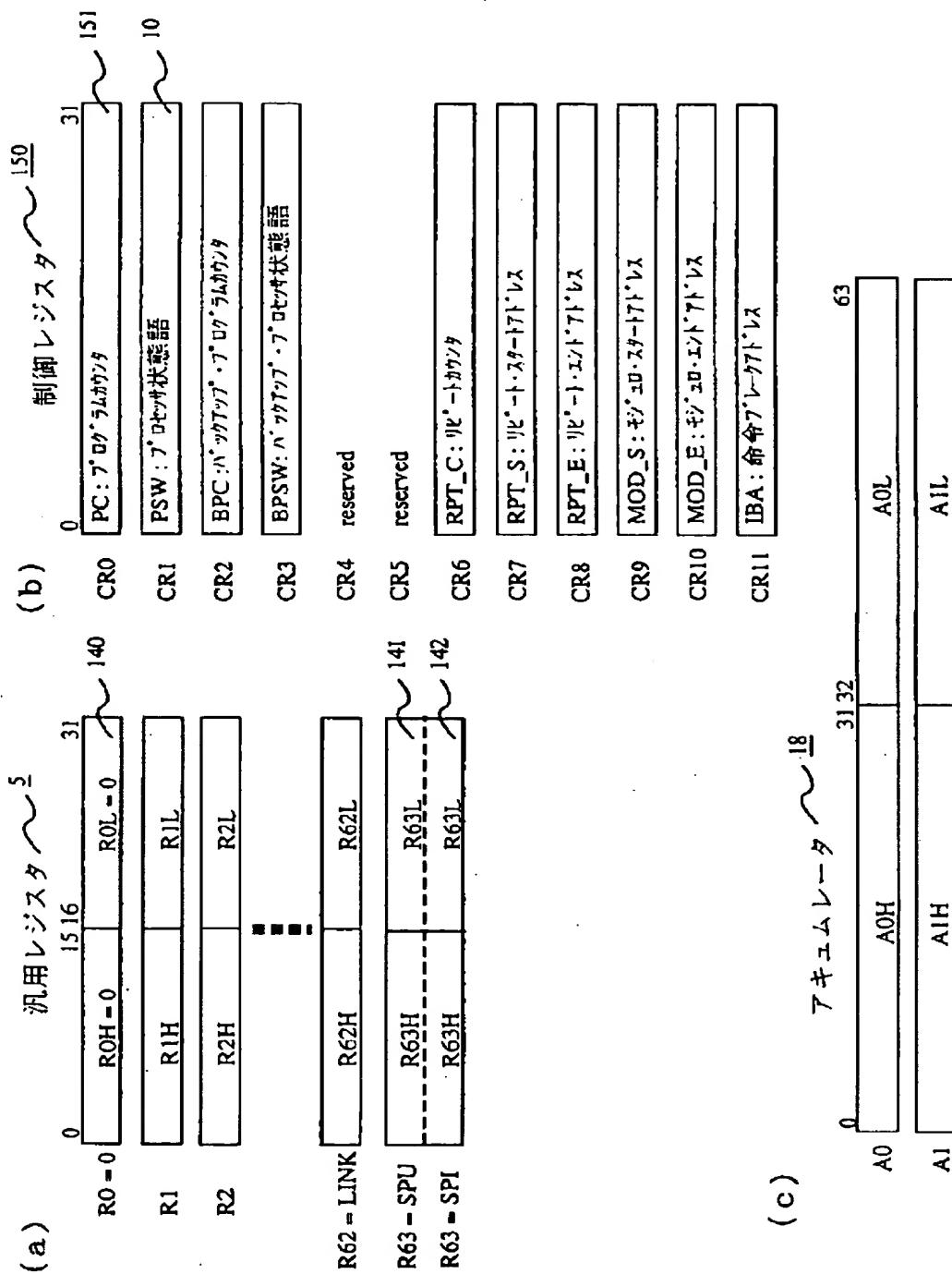
【図6】

メモリユニットでの実行	191	192	193	194		190
整数演算ユニットでの実行	IF	D/A	E/M	W		195
メモリユニットでの実行	IP	D/A	E/M	W		
整数演算ユニットでの実行	IP	D/A	E/M	W		
メモリユニットでの実行	IP	D/A	E/M	W		
整数演算ユニットでの実行	IP	D/A	E/M	W		

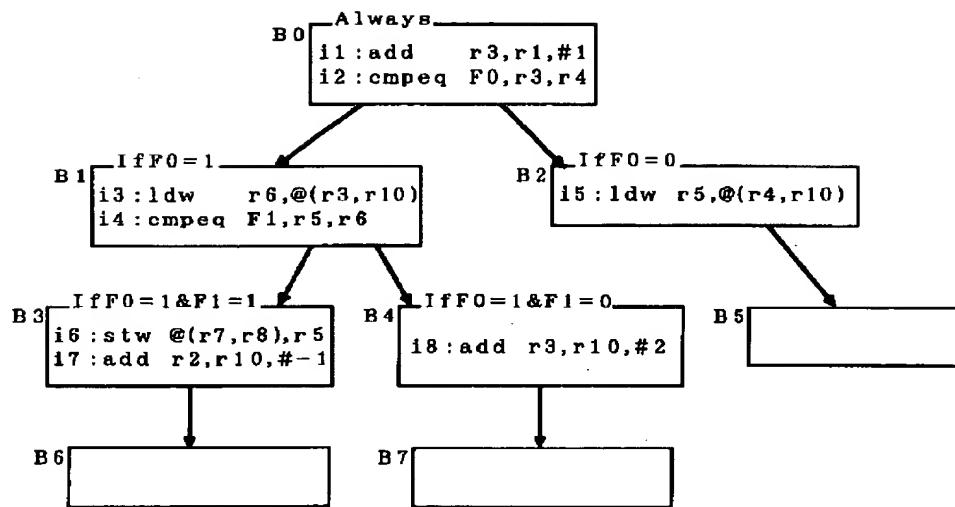
【図7】

メモリユニットまたは 整数演算ユニットでの実行	IP	D/A	E/M	W		200
メモリユニットまたは 整数演算ユニットでの実行	IP	D/A	E/M	W		
メモリユニットまたは 整数演算ユニットでの実行	IP	D/A	E/M	W		

【図4】



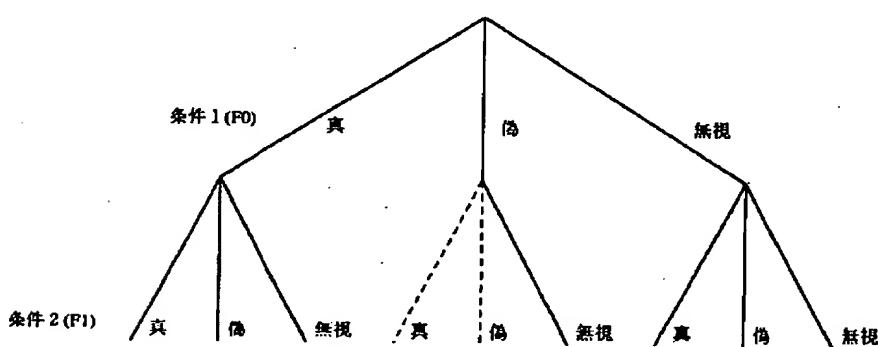
【図8】



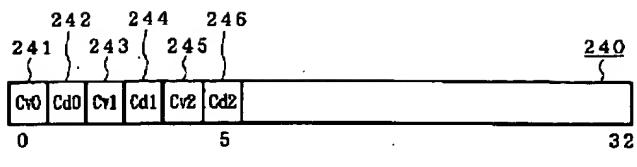
【図10】

CC	F0	F1	F2	CC	F0	F1	F2	CC	F0	F1	F2
00000	真	真	真	01001	偽	真	真	10010	無視	真	真
00001	真	真	偽	01010	偽	真	偽	10011	無視	真	偽
00010	真	真	無視	01011	偽	真	無視	10100	無視	真	無視
00011	真	偽	真	01100	偽	偽	真	10101	無視	偽	真
00100	真	偽	偽	01101	偽	偽	偽	10110	無視	偽	偽
00101	真	偽	無視	01110	偽	偽	無視	10111	無視	偽	無視
00110	真	無視	真	01111	偽	無視	真	11000	無視	無視	真
00111	真	無視	偽	10000	偽	無視	偽	11001	無視	無視	偽
01000	真	無視	無視	10001	偽	無視	無視	11010	無視	無視	無視

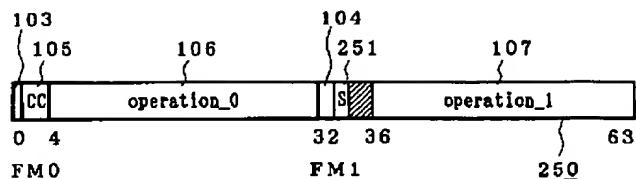
【図11】



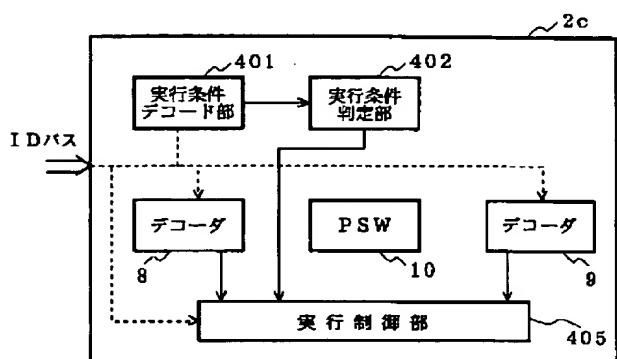
【図13】



【図14】



【図15】



【図17】

